UFRMI L1	*Programmation Python* 2018 *Examen session* *Durée 1H30* 1

EXERCICE 1

Soit l algorithme d'approximation de racine selon Héron d'Alexandrie:

```
algorithme carrel
variable
     a,a0,an: reel
     i:entier
debut
     ecrire('calcul de racine carrée de :')
     lire(a) // le nombre dont on cherche la racine carrée
     ecrire( 'le carré le plus proche')
     lire(a0)
     pour i<--1 jusqu'à 5 faire
          an<--(1/2)*(a0+a/a0)
          ecrire(an)
          a0<--an
     finpour
     ecrire(an)
fin
```

1.1Traduire l'algorithme en python.

1.2 Donner la trace et le résultat d'exécution pour 9 et 15: (faites au moins 5 itérations de la boucle en suivant les valeurs de i, a0, a et an)

1.3 Ecrire une version plus simple du programme en python en utilisant 'math'

1.4 Transformer le programme (2.2) en une fonction python *def carrel(n) : .....return an*

1.5 Ecrire un programme principal permettant de calculer la différence entre la valeur approchée et la valeur réelle.

EXERCICE 2

The Gregory–Leibniz series

$$\pi = 4\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1} = 4\left(\frac{1}{1}-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+-\cdots\right) = \cfrac{4}{1+\cfrac{1^2}{2+\cfrac{3^2}{2+\cfrac{5^2}{2+\ddots}}}}$$

2.1Ecrire un programme permettant d'afficher pour n allant de 1 jusqu'à 10 les valeurs $(-1)^n$

2.2 Utiliser la formule de Leibniz pour écrire un programme itératif en python qui permettra d'avoir une valeur approchée de Pi.

*Indications :from math import * : à faire avant toute utilisation de fonction mathématique telles que : cos (n), abs(n), sin(n), sqrt(t), math.pow(x,y) Retourne x élevé à la puissance y.*

VALIDEUR LMD conseils-sujets d'examens-corrections

UNIVERSITE FHB COCODY	
UFRMI	2019
QCM( )=1 0 F=-1)	**Examen :** Programmation
Rendre la première feuille	**Niveau :** Licence 1
	Session : 1
	Durée : 1H30min

Q1- Pour produire un fichier « .py : » , on utilise un
- (a) Un éditeur
- (b) Un exécuteur
- (c) Un compilateur
- (d) Un traducteur

Q2- 2. python est un langage développée par
- (a) Hewlett-Packard
- (b) Steve Jobs
- (c) John von Neumann
- (d) Guido van Rossum
- (e) Bill gates

Q3 – L'instruction suivante , t= [4,6,5,8,7], crée :
- (a) un ensemble t
- (b) une itération t
- (c) une liste t
- (d) une table

Q4 – une fonction écrite en python contient obligatoirement :
- (a) une itération
- (b) le mot clé 'function'
- (c) le mot clé return
- (d) une fonction procédure
- (e) end_fonction

Q5 - Que va afficher à la console le programme ci- dessous ?

```
x = int( 1)
if x>1 :
x = x+2
print(x)
```

- (a) 0
- (b) 1
- (c) 2
- (d) 3

Q6- Lequel des mots réservés de python permet de forcer la sortie d'une boucle :

```
for i in range(10) :
    ...
```

- (a) try .. catch
- (b) break
- (c) exception
- (d) exit

Q7 Soit la fonction

```
def f ( x ) :
    ....
```

Comment peut-on l'appeler dans un programme principal?
- (a) f(int x)
- (b) f(a)
- (c) f(a,b)
- (d) f(10)

Q8

```
a=2
b=5
if a>8:
   b=10
elif a>6:
   b=3
```

Que vaut la valeur finale de la variable b ?
- (a) Je ne sais pas
- (b) 3
- (c) 5
- (d) 10

VALIDEUR LMD

EXERCICE 1 Soit le programme en python ci dessous :

```
def MaFonction(n): # définition de fonction
      res = []
      k = n
      iter = 0
      while k > 0:
            iter = iter + 1
            res.append(k%2)
            k = k//2
      return res
#programme principal
tb=MaFonction(16)
print tb
```

1°) Exécuter le programme pour 16, 4 et 9.
2°) En écrivant les tableaux résultats de la dernière cellule à la première, trouver ce que fait le programme.
3°) Ecrire plus simplement le programme en utilisant les fonctions prédéfinies : bin, hex, int...

EXERCICE 2

Soit l'algorithme de Héron d'Alexandrie permettant de calculer une valeur approchée de racine carrée de x

```
Algorithme carel
variable
   A,x0, xn: reel
   i:entier
debut
   Ecrire('calcul de racine de :')
   lire(A)
   Ecrire('saisir la racine carre
   la plus proche!')
   lire(x0)
   pour i← 1 à 5 faire
      xn ← (1/2)*(x0+(A/x0))
      Ecrire(xn)
      x0←xn
   finpour
Fin// algorithme
```

```
C:\Documents and Settings\Administrateur\Mes d
calcul de racine de :
43
saisir la carre la plus proche 28
28.7750000000
28.7685445247
28.7685394928
28.7685394928
28.7685394928
```

1°) Traduire l'algorithme en python.
2°) Transformer le programme en une fonction en retirant l'ordre d'écriture de xn.
3°) en utilisant la fonction *sqrt(x)* de la bibliothèque *math* et la fonction écrite à la question précédente, écrire un programme qui permet d'afficher après 5 itérations l'écart entres la valeur trouvée par la méthode de Héron d'alexandrie et la fonction *sqrt(x)*

AIDEUR LMD

UFRMI L1	*Programmation Python* 2017 *Examen session1* *Durée 1H00*

EXERCICE 1: *Questions de cours*

11 Parmi ces valeurs lesquelles ne sont pas des « ENTIER » : 1 -43 21.0 4E-06 5C

12 Traduire en python l'algorithme et donner le résultat de la ligne 11

```
1 ALGORITHME exemple5bis
2     VARIABLE a, b, c : reel
3 FONCTION Produit (x, y  : reel): reel { parametres }
4     DEBUT
5           retourner( x * y)
6     FIN
7 DEBUT
8     Ecrire ('a b ? ')
9     Lire (a, 'b')
10    c ← Produit (a, b)
11    ecrire(Produit (5, 10))
12    ecrire ('c = ',c);
13 FIN
```

13- Donner pour chaque donnée le type correspondant en python :
1.0
4
21.14192588
3.14 E+01
-100
[1, 12, 45, 66, 49, 64, 51]
« False »

14 for i in range (1,10) :
    print (i)
affiche quelle(s) valeurs de i ? justifier

15 Qu'est ce que le trans-typage ? Donner un exemple utilisé pendant la saisie des données en python.

IA IDEUR LMD

EXERCICE 2

```
1 def MaFonction(n): # definition de fonction
2       res = [ ]
3       k = n
4       iter = 0
5       while k > 0:
6             iter = iter + 1
7             res.append(k%2)
8             k = k//2
9       return res
10 #programme principal
11 tb=MaFonction(16)
12 print (tb)
```

2.1 Donner l'explication de chaque ligne
2.2 Donner la trace et le résultat d'exécution pour 4 ; 9 et 16.
2.3 En déduire ce que fait l'algorithme.
2.4 Les fonctions 'bin' et hex permettent de faire plus simple, Ecrire un programme qui demande un nombre et le convertit en binaire et en hexadécimal

IR LMD